

Mobile App Design for Teaching and Learning: Educators' Experiences in an Online Graduate Course



Yu-Chang Hsu and Yu-Hui Ching
Boise State University, USA

Abstract

This research explored how educators with limited programming experiences learned to design mobile apps through peer support and instructor guidance. Educators were positive about the sense of community in this online course. They also considered App Inventor a great web-based visual programming tool for developing useful and fully functioning mobile apps. They had great sense of empowerment through developing unique apps by using App Inventor. They felt their own design work and creative problem solving were inspired by the customized mobile apps shared by peers. The learning activities, including sharing customized apps, providing peer feedback, composing design proposals, and keeping design journals (blogging), complemented each other to support a positive sense of community and form a strong virtual community of learning mobile app design. This study helped reveal the educational value of mobile app design activities and the web-based visual programming tool, and the possibility of teaching/learning mobile app design online. The findings can also encourage educators to explore and experiment on the potential of incorporating these design learning activities in their respective settings, and to develop mobile apps for their diverse needs in teaching and learning.

Keywords: Online learning; mobile app design; programming; App Inventor (AI); virtual learning community (VLC)

Introduction

The rapidly advancing mobile computing technologies along with abundant mobile software applications (“mobile apps” hereafter) make ubiquitous mobile learning possible (Johnson, Adams, & Cummins, 2012). The major affordances of mobile computing technologies for learning include user mobility resulting from device portability, relatively strong computing power in small devices, and always-on connectivity (Hsu & Ching, 2012). These affordances lead to tremendous potential for innovative uses of mobile technologies in education. In addition, mobile devices such as smartphones or iPod touches are vastly gaining popularity (Johnson, Levine, Smith, & Stone, 2010) due to the availability of various easy-to-use mobile software applications. There were more than 800,000 mobile apps for iOS devices in Apple’s AppStore and Android OS devices on Google Play (formerly Android Market) respectively across over 20 categories (Wikipedia, 2013a; Wikipedia, 2013b).

The innovation in mobile apps has raised interests among educators because it facilitates teaching and learning (Johnson et al., 2012). However, relevant and appropriate apps are not always available; teachers can only judge apps by reviews or anecdotes instead of testing them. These reasons lead to the needs and benefits for creating one’s own mobile apps for teaching and learning. In addition, empowering educators to create their own mobile apps for teaching and learning is likely to lead to advantageous use of mobile technology that benefits student learning. For example, pre-service science teachers engaged in the development and customization of a smartphone application that supported student scientific inquiry (Price et al., 2012). However, designing educational mobile applications remains a challenge for educators without previous programming experiences. In particular, the effort and time required to learn to code in textual programming languages make it unrealistic for educators to develop their own apps. Fortunately, some visual programming languages have been developed to address these challenges. App Inventor (AI), a web-based GUI (graphical user interface) builder (Wolber, 2010) by Google Inc. and Massachusetts Institute of Technology, allows anyone with a Google account to prototype and develop Android mobile apps through its visual block-based programming language by drag-and-drop actions (Hsu, Rice, & Dawley, 2012).

Currently, most examples using AI for education focus on teaching programming to K-16 students with various levels of prior programming knowledge (e.g., Margulieux, Guzdial, & Catrambone, 2012; Morelli, Lake, Limardo, & Tamotsu, 2011; Wolber, 2010; Wolber, 2011) (see <http://appinventor.mit.edu/explore/stories.html>). In addition, most of these current practices and studies were conducted in face-to-face settings. While these research findings and stories revealed success in using AI for teaching and motivating students to program, there is little practice or research on examining non-programmer adults’ online learning experiences of creating their own mobile apps for education. This paper aims to help fill the void of research on online learning of mobile app programming of educators who are neither programmers nor computer science majors. By examining the experiences of non-programmer educators in mobile

programming in online environments, we hope the findings of this research help elucidate the potential and value of promoting mobile app programming for authentic educational uses.

Literature Review

Teaching and Learning Programming Online

Among the scarce studies on teaching programming online, Wang (2011) reported her experiences teaching object-oriented programming online to students majoring in computer science. She argued that it is difficult for students to get instant instructor feedback in online environments, while this type of feedback is especially important for programming learners. Also, she indicated it is critical for students to see how others approach the same programming problem with their solutions and collaborate with other students. Both aforementioned aspects can be challenging for instructors and students. Based on her research, she suggested three strategies to improve students' online learning experiences of programming: 1) creating a virtual computer lab to allow students to engage in programming activities immediately without the frustration and constraints of installing programming environment software at home or in the workplace; 2) adding more multimedia materials into the course such as videos that make the instruction engaging and easy to follow; 3) creating a sense of community among students to help them support each other. In a course on Java programming, McKelvey and Curran (2012) incorporated discussion forums in a course management system to facilitate discussions on assigned programming topics, team interaction, and communication. Considering the suggestions from Wang (2011) and McKelvey and Curran (2012), we determined that it is important to build an online community to support online programming activities.

Online Learning Community to Support Learning of Programming

Learning mobile programming can be challenging for educators who are not programmers, especially in an online learning environment. As such, an effective learning environment should build in rich support for these educators. In the course of mobile app design examined in this current study, an online learning community was formed to support learners. A learning community refers to a social community of learners who share knowledge, values, and goals while learning (Rovai, 2002b). There are four critical social components in an online learning community: communication, collaboration, interaction, and participation (Lock, 2002). Through active participation, learners collectively inquire into specific topics, share and exchange thoughts and experiences, and make improvements to ideas to develop deeper understanding (Lave &

Wenger, 1991). Support from peers in the community can greatly enhance learning in an online environment. In addition, members in a well-functioning community should develop a strong “sense of community” as demonstrated by the mutual interdependence among members, connectedness, trust, interactivity, and shared values and goals (Rovai, 2002b, 2002a). Various instructional strategies were adopted in the current study to help build a strong online learning community and to strengthen the “sense of community” among learners.

The Benefits of Visual Programming Tools for Non-Programmers and Using Mobile Apps as End Products

Visual programming can be a good solution to help non-programmers learn programming more easily. Visual programming tools enable people to see and test what they build immediately after putting together the pieces of different components. These tools also create a more enjoyable programming experience by reducing the frustration of getting lost in textual codes and debugging. Since Glinert's (1986) pioneering work on BLOX (a visual programming language consisting of puzzle-like pieces), there have been a few successful visual programming tools available. For example, Scratch is a free tool that makes it easy to create one's own interactive stories, animations, games, music, and art in two-dimension format (Lifelong Kindergarten Group, 2006). Another tool, Alice, is a three-dimensional (3D) programming environment for creating story-telling animations, playing interactive games, or sharing videos on the Web (Carnegie Mellon University, 2008). One great advantage of introducing programming to novices with visual programming languages is that it can help them avoid syntax errors commonly seen in working with textual programming languages. In addition, the drawer analogy used for arranging the puzzle pieces (called “blocks” in App Inventor) with similar function can reduce the need for novices to remember exact textual codes (Turbak et al., 2012), which can greatly reduce the potential cognitive load caused by programming with textual codes (Margulieux, Guzdial, & Catrambone, 2012).

App Inventor (AI) also features drag-and-drop visual programming, which lets designers see how different pieces come together, and how their programming relates to the behaviors of their artifacts/products—the mobile apps (Hsu, Rice, & Dawley, 2012). AI is a free web-based tool that consists of two major elements: Component Designer (see Figure 1) and Block Editor (see Figure 2) which together allow users to develop mobile apps running on Android devices. Component Designer lets one design the app's interface and integrate non-visible components (i.e., feature/function not visible to users on the mobile device interface) such as GPS (global positioning system) or sound. Block Editor allows one to program mobile apps' behaviors and to control how apps react under certain circumstances. This tool has great potential for enabling educators with limited programming knowledge and experiences to experiment and design mobile apps that suit their professional needs.



Figure 1. Component Designer.

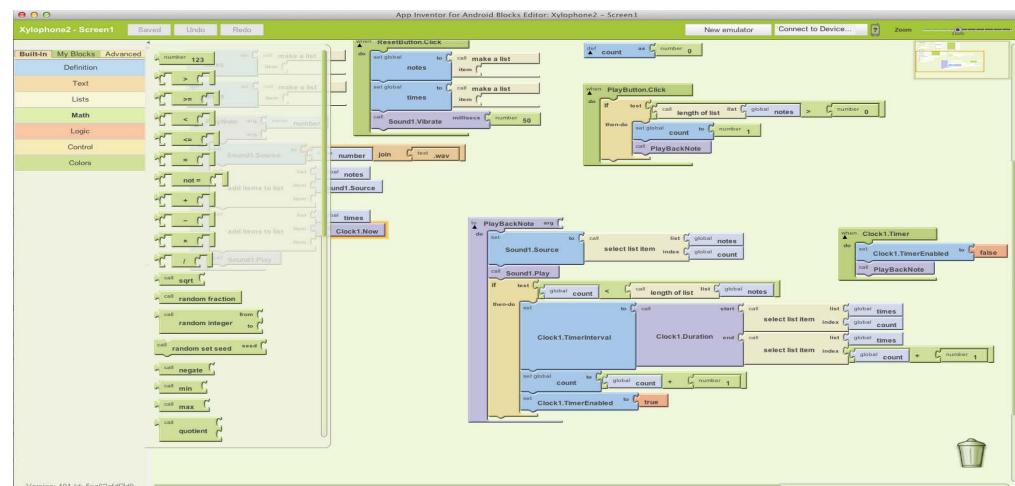


Figure 2. Block editor.

Another strength of AI lies in the design products themselves—the mobile apps can be tested, used, or played immediately on one's mobile device. This gives a sense of practicality and reality in achievement, which can be very satisfying experiences in one's learning. Like Scratch and Alice, the process of creation with AI can stimulate fun, creativity, and learning about programming. AI also takes it further—the products created through AI can go anywhere with the users and afford practical use in real-life.

Research Purpose and Questions

The purpose of this research is to explore how non-programmer adults who are educators learned to design mobile apps for teaching and learning through peer support and instructor guidance in an online graduate course. The research questions are as follows:

- What is students' sense of community in this online course?
- What are students' overall experiences of learning mobile programming in an online course?
- How does students' sharing of mobile app ideas, app customizations, and peer feedback influence their learning of mobile app design?
- What are students' perceptions of major assignments to support their app design experiences in this course?
- What are students' perceptions of App Inventor as a tool for learning mobile app design and for developing mobile apps?

Through this study, the authors aim to 1) depict non-programmers' experiences of learning mobile app design in an online course; 2) explore the practices and challenges in teaching mobile app programming online; 3) examine the strengths and constraints of App Inventor as a web-based tool for learning and teaching mobile app programming; 4) provide useful design suggestions for educators interested in incorporating mobile app design in developing students' creative problem-solving through programming.

Methods

Context and Learning Environments

This study was implemented in a fully online graduate course on mobile app design in a mid-size state university in the northwestern United States. The goal of this course was for students to learn to design engaging and practical Android mobile apps for teaching and learning by using App Inventor. This course consisted of the following components:

- 1) Moodle learning management system, where course instruction was posted, and students shared their app design and provided each other constructive feedback on weekly discussion forums;

- 2) individual app design journal, where on their blogs students reflected upon their design experiences using AI, exploring learning resources, and their inspiration from peer ideas and projects;
- 3) a Google site where students posted their app files and created QR (quick response) codes of their apps for review and testing.

Through the various components, this course intended to help sustain a virtual learning community (VLC) for non-programmer adults to support and inspire each other's learning and design work, and cultivate a sense of community, including the feelings of connectedness among community members and commonality of learning expectations and goals, which is critical to successful learning in a VLC (Hsu & Ching, 2011). On weekly forums, the sharing of mobile app design and providing peer feedback helped students engage in communication, interaction, and participation for collaborative knowledge construction in a VLC (Lock, 2002). Through keeping a design journal, the students engaged in reflective practice and internal conversation for individual knowledge creation that enhanced their learning (Baker, Kolb, & Jensen, 2002; Shumack, 2010). Journaling in the form of blogging has been commonly used for reflective practices in learning (e.g., Ferdig & Trammel, 2004; Ladyshewsky & Gardner, 2008; Shoffner, 2009). The thirteen students enrolled in this course included K-12 teachers, school technology specialists, and corporate personnel from education industries. Although AI provided an emulator that allowed testing an app virtually, students were required to own or at least have access to smartphones or tablets running Android OS version 1.6 or higher so that they could be engaged in the full experience of app design and testing, because some functions (e.g., shaking the device to activate an action) were only available on physical mobile devices. The learning environments and components, including AI, complemented each other to help form a VLC of mobile app design (see Figure 3).

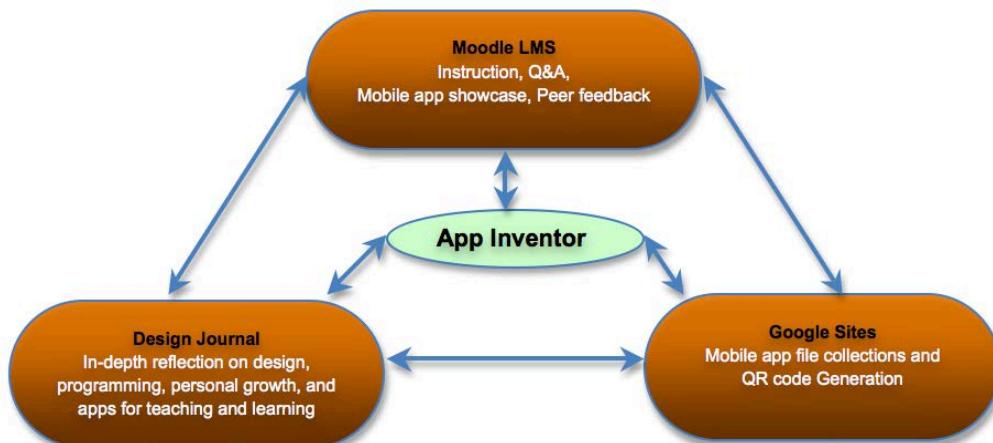


Figure 3. The integration of learning environments in the online course in mobile app design.

Mobile App Design Activities and Reflective Practice

Each student learned by doing (designing)—they developed eight practice apps by following the tutorials in the textbook authored by Wober, Abelson, Spertus, and Looney (2011), and incorporated customizations on interface, theme, and function into their apps from Week 3 to Week 10. The students then developed their own app design proposals individually using the templates provided by the instructor. The proposal included three major sections:

- 1) a planning table that involved user analysis, context analysis, rationale for the app, and future plan with this app;
- 2) app component map, where the students described and depicted the interface components and how users would interact with their apps;
- 3) app behavior flowchart, where the students presented the logic of app behaviors in detail and the thought in terms of blocks programming and how the app worked behind the scenes.

At the end of the semester, each student completed his/her final project by submitting one final original app (see Figure 4) based on the ideas delineated in their app design proposal. Throughout the semesters, students provided each other feedback regarding the customized apps, ideas for further customizations, design proposals, and final project apps. Students also shared resources such as web tutorials, and helped answer questions on app programming and debugging. In addition to the actual design (learning by doing) and peer review (social learning) activities, students also engaged in reflective practice by keeping app design journals on their blogs dedicated to this mobile app design course. Journal writing has been established as an experiential approach consisting of internal conversation for knowledge creation and facilitating one's learning (Baker, Kolb, & Jensen, 2002; Shumack, 2010). Through design journals, students in this course documented and reflected on their growth as app designers. They reflected upon the joys and struggles encountered during the design process, the inspiration from their peers' work, ideas for future apps and customizations, and what the learning experiences meant to them in their professional contexts. In sum, the learning environments, hands-on app programming activities, app design proposals, peer review of the apps, and reflective practice complemented each other to enrich and support students' learning of mobile app design in an online setting. Figure 4 shows a variety of student final project apps that include app genres such as utility, learning, and game.



Figure 4. Various types of students' project apps.

Research Design and Data Sources

This study applied a mixed-method design. The researchers collected both quantitative and qualitative data to depict students' mobile app design activities in this online course. A web-based survey was implemented at the end of this course (Week 16), which collected students' demographic information, perceptions of sense of community, reporting of time on participation in class activities, perceptions of learning activities, and perceptions of App Inventor as a tool to create mobile apps. The survey was conducted by the second author and the participants' identification information was removed by the second author before data analysis, so the first author (also the course instructor) could not link participants' responses with their names. This procedure was in place to help avoid potential issues of research participation coercion and response bias. Also, students' artifacts, which included customized practice apps and final project apps, were examined to reveal their learning growth. In addition, students' 10 weeks of design journal entries were analyzed to help triangulate and interpret students' responses in the survey. The rich data and analysis helped us answer the research questions mentioned above, including perceptions of learning supported by a virtual

community, experiences in the various learning activities designed to help learning app design, and influence of peer feedback and app ideas on students' final projects.

Results and Discussion

Participants, Mobile Devices, and Android Emulator

Ten of the thirteen students enrolled in this course participated in this study. All of the 10 participants own the Android phones they used to develop mobile apps. The lead brands owned by participants were HTC (5 students), followed by Motorola (2 students), LG (2 students), and Samsung (1 student). In terms of Android OS version, six used Android 2.3 and its variations, two used Android 2.2 and its variations, and one used Android 2.1. One participant did not specify the Android OS on his/her phone. Two of the participants also indicated they owned tablets during the class, and one of them also tested his apps on tablets in addition to testing on his smartphone, before sharing with peers. In addition to testing apps on mobile devices, four of the ten participants indicated they tested apps using the Android Emulator (a virtual mobile device) that came with Android SDK (software development kit), while five of the ten participants rarely used it and one never used it.

Time on Task

The majority of students (7 of 10) spent between 2 to 4 hours each week creating original practice apps by following the tutorials in the textbook. For customization of the practice apps, six students spent about 4 hours each week. Eight of ten students spent 1 to 2 hours searching for or studying tutorials online to help their app customizations. Most students (8 of 10) spent one hour reviewing peers' apps each week. While 6 of 10 students spent 4 to 5 hours developing their app design proposals, one student spent as little as 2 hours on the proposal and another spent as many as 25 hours on this task. Time spent on the final project app varied to a great extent, from 10 hours to more than 80 hours.

Sense of Community

For each item on the Sense of Community scale (see Rovai, 2002a, 2002b), the possible score ranged from 1 to 4 points (1 = *strongly disagree*; 2 = *disagree*; 3 = *agree*; 4 = *strongly agree*). An examination of the item means showed that the participants' responses mostly fell into the "agree" category (near or above 3) (see Table 1). The mean score across the 20 items of the SoC scale is 3.13, which indicates students had an overall positive sense of community in this app-design course. When further breaking down the SoC scale into the subscales, the mean score across the 10 items of the Connectedness subscale was 2.99, which indicated students felt connected to class

members. In addition, the mean score across 10 items of the Learning subscale was 3.26, which showed students felt positive about their learning experiences in this course.

Table 1

Survey Item Response Mean Scores

Item no.	Items	Response mean score
1	Classmates care about each other	3.10
2	I am encouraged to ask questions	3.30
3	Feel connected to others	3.10
4	Feel easy to get help	2.90
5	Feel a spirit of community	3.20
6	Feel receiving timely feedback	3.30
7	Feel the course being like a family	2.50
8	Feel easy exposing gap in my own understanding	3.10
9	Feel not isolated in this course	3.20
10	Feel willing to speak/communicate openly	3.20
11	Trust others in this course	3.20
12	Feel this course resulting in more than modest learning	3.30
13	Feel able to rely on others in this course.	3.10
14	Feel other students helping me learn	3.30
15	Feel members of this course depending on me	2.30
16	Feel being given ample opportunities to learn	3.20
17	Feel certain about others in this course	3.10
18	Feel my educational needs being met	3.40
19	Feel confident that others will support me	3.10
20	Feel this course promoting a desire to learn	3.60

Note. Boldface: "Connectedness" subscale; non-boldface: "Learning" subscale.

Students' Perceptions of Overall Experiences of Learning Mobile Programming in an Online Course

In the following subsections, we present the themes that reflect different aspects of students' mobile app design experiences emerging from participants' open-ended responses and their blog entries.

Overall app design experience in this course.

Students spoke highly about their overall experiences in this class on several aspects. One student indicated:

It was so much fun to be creative and imagine the possibilities of using a program like this with students, and to develop apps for the classroom. Although the class was challenging, I felt like I could manage the work with the support from both my peers in the class and

[the instructor]. What a great feeling to have a completed, original app developed.

Another student commented how immersed and engaged he/she was when designing apps:

I really enjoyed the time that I had to create the app and wish that I had all the time in the world to develop more.

...I have so much fun doing it that about 9 hours went by before I completed the app, and all weekend I have thought about 10 more things that I would like to change to make it better.

Learning transfer and beyond.

One student with other experiences involving computational thinking and logic revealed how he/she benefited from this course: «[it] influenced my understanding of logic and programming in other areas, especially in Excel and MySQL statements and formulas.»

Yet another student took the learning experiences beyond this course and created wonderful teaching moments for his/her students:

I have taken the apps I have created so far and let some of my students try them out. This past week, after trying the PaintPot app, one of my students asked, “Are we going to get to use these in class?” The level of excitement on her face was priceless. So, I opened a dialogue, asking how she thought we might be able to use apps in class. We shared an open dialogue that included many great ideas and platforms from which I can begin to shape apps in the future.

Growth as an app designer.

Some students found it fun and fulfilling and talked about their growth as app designers:

I'm finding it fun to begin to see the secret door behind app developing. I'd only been a previous user of apps before this class, and had never thought about the complexities that go into the design and programming of their functions.

Around the time when students were building the last three of their eight practice apps, most students became fluent with the block-based programming language: «I'm beginning to understand some of the logic behind the different options and blocks, something I never have before in previous programming exposure. »

Empowerment.

Many students felt a sense of empowerment and achievement because they could handle a tool that allowed them to use their creativity and develop something that might not have existed before. For example,

I feel more empowered by being able to develop an idea and actually make it real, and have others run my app. There is a strong feeling of success. Also, knowing I now have the ability to make something that may not exist yet is a great feeling.

I do feel empowered by learning to use the blocks to program behavior, creating a working original app, and sharing my work with my peers and also with my colleagues at work.

I definitely feel empowered and have actually started to incorporate App Inventor into my Introduction to Java Programming course at my high school. All the knowledge of this course is completely new to me and very useful.

Perceptions of Class Activities

Learning by doing: App design and customization.

Students were very positive about the learning-by-doing experiences provided by the actual mobile app design and required customizations. They also appreciated the opportunities of making mistakes and resolving the problems during the processes.

This is a great way to use authentic learning for the learners- give them a real practical and useful goal, then allow them the chance to make mistakes and search out the answers for themselves. Very cool.

App design proposal.

Some students recognized the benefits of designing their apps by starting with a proposal with appropriate prompts:

This is pretty much the perfect way to begin designing an app. The Template caused me to really think about how and why a user would want to use the app. It also forced me to think ahead a few steps and consider future modifications.

I think this is an absolutely necessary step. By the time I started my actual design I already had most of my graphics done due to the design proposal stage.

Others found it authentic to build an app by figuring out the blueprint first, but hoped for more time on this stage:

I think there are many benefits to creating the proposal, it adds a layer of realness to the process. All programs start on paper, and build up from there. The challenges in the process were due to time. I think there should have been more time to create the design. Just felt rushed through the process.

App design journal (blogging).

The app design journal (see Figure 5 for an example) helped students document their work, and reflect on their growth as well as the inspiration obtained from peers' work. Students also appreciated that the journal can be incorporated into their degree program portfolios.

Keeping a journal allows me to express my thoughts and opinions about the weekly process more than just the design, which is beneficial for me as a professional.

This is a great journal to refer back to and having this already in place gives me a leg up on preparing my portfolio. I wish all courses made this a requirement.

Keeping the journal is also beneficial for developing students' own apps: « This week, we began design of our final app for the course. I looked back upon my previous blog reflections to help generate some ideas. »

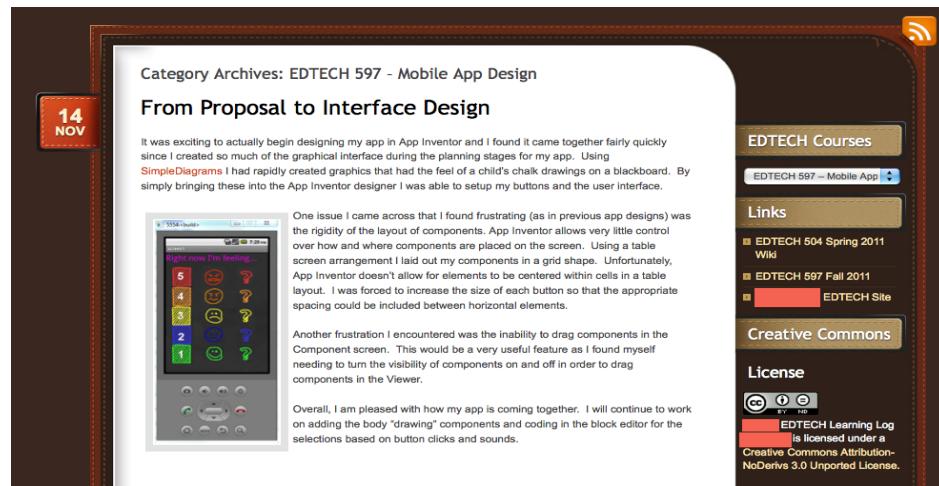


Figure 5. An example of app design journal.

Sharing Is Good: Mobile App Ideas, Customizations, and Peer Feedback in the Virtual Community

Some students were not quite sure about openly sharing their ideas and design initially as that might expose the gap in their knowledge and skills. However, the concern and uneasiness was relieved after realizing the benefits of openly sharing in an environment where they felt safe and comfortable.

I was initially nervous about sharing everything but then I realized we are all in the same boat. Everyone was so willing to help one another. There was a great sense of community and I learned a huge amount by looking at my peers' ideas and blocks.

Students found it useful to review others' work for inspiration and to get feedback:

I like reviewing the designs of others to get ideas and help troubleshoot. I like receiving honest feedback about my own apps, often to catch things I may miss.

I like it a lot, the feedback is great and it has been tremendously inspirational to see the innovation and talent that exists among my peers.

The sharing activity also provided examples and modeling of creative design problem-solving:

Without seeing the work of my peers I wouldn't have been exposed to the numerous approaches to customization of these projects.

Benefit was huge—learn about different possible ways to achieve the same programming outcomes

Experiences of Designing Mobile Apps with App Inventor

Strengths.

Students appreciated the accessibility of AI as an app design tool: « It's a great, no-fear approach to programming – people are normally turned off by foreign programming language. »

In addition, students considered this course led to fun learning experiences that they enjoyed sharing with others: « It was a lot of fun and sparked quite a bit of interest with people when I talked about the course. »

One student specifically commented on the educational value of app design with AI:

I would introduce this class to a computer class or a class with my students. I think that this type of activity is very important for students to learn. Building these apps and customizing them forced us to use creative thinking to solve design problems. This is where we want our students to go.

AI as a tool for learning app design and programming apps.

Overall, students are very positive about AI as a learning tool and design tool. The students like how quick and easy AI allows for putting together a working app:

App Inventor's power is in how quickly a functional app can be developed. I can see myself using it as a prototyping application to see quickly how an app would look and function, then use a more advanced development environment to build the app with more functionality.

Considering the convenience of AI for programming app behaviors, testing apps, and documenting design thinking, one student commented: « ...you could activate or deactivate blocks. I also liked that I could add comments to my blocks. »

On a broader scope, one student commented on the available mature community external to this course for developing mobile apps with AI. This kind of support can help sustain interests in developing apps with AI:

I like ... there is an informed and supportive community of users on the Internet. I was able to solve a lot of problems by browsing through message boards.

Room for improvement.

While students are excited and passionate about developing mobile apps with AI, it can be challenging at times. For example, in terms of the block-based interface, one student indicated that « The brick [block] designer is annoying once your app begins to grow larger. »

This is a fair observation because a complicated app can involve many blocks. Although one can collapse the view to minimize the space taken by each group of blocks, it is difficult to get a full-view of all blocks when one needs to skim through and identify problems—it requires much scrolling vertically and horizontally on the page/screen. Another limitation of AI, as one student mentioned, is the constraint of layout of interface components—AI allows very little control over how and where components are placed on the development screen.

In addition, students who are more advanced hoped for the ability to edit the Java (i.e., the original programming language used for developing Android mobile apps) text code directly: « You are not able to toggle between blocks view and the actual code. I think it would be a very useful way to learn code design. »

This comment is particularly pertinent to advanced app developers who want to develop highly customized apps. It is also relevant to educators interested in teaching programming and using AI as a stepping-stone for teaching advanced textual programming languages. Although AI is a sophisticated and powerful web-based program that allows people to build complicated apps, developers who use AI need to work within the provided blocks and features offered by the AI environment. One tool called App Inventor Java Bridge does allow developers to import App Inventor components into the authentic Android SDK (software development kit) environment of text codes, but it is not built into the App Inventor environment and requires extra effort to make the transition.

Instructional Design Implications and Suggestions

The integration of multiple online learning environments and learning activities was overall positively recognized by the students in this class. The students also appreciated AI as a great tool to help them develop a functioning app and develop programming logic. The sense of achievement was also high when apps were completed. Together, these various components helped support non-programmer adults learn mobile app design through visual programming in a virtual learning community, where they contributed to each other's learning growth as mobile app designers. However, students' learning experiences can still be improved by factoring in their feedback and challenges encountered in this class.

More Time for Developing App Design Ideas and Proposal

Some students felt they needed more time on the design proposal, and one week was a bit rushed. While in this course students continued developing ideas and tweaking their design with one week designated for interface design and another week for working on the blocks, the instructor should consider arranging students' submissions of revised proposals to allow them to think through and incorporate peers' feedback. Giving more time for planning on paper could also help students reduce potential obstacles to programming app behaviors in blocks. The proposal was scheduled at the 11th week of the 16-week semester; however, instructors could assign the first draft due earlier (e.g., around the 6th week, one-third of the semester), and the second draft due around the 11th week (two-thirds of the semester), which would allow students to update their plans as they became more advanced in app design skills. The two-stage method would allow students to reflect on their growth in learning and changes of ideas connected to app design journal activities.

Design Journal (Blogging)

While many students found it beneficial to keep a design journal through blogging, some students found it overlapped with the weekly online discussion forum activities. Instructors might want to provide more specific instruction/suggestions on what to focus on in respective assignments. For example, in forum discussions (where students shared their completed app customizations), students can focus on describing and critiquing the technical aspects, or problems encountered during their customizations. In the app design journal, instructors might want to be specific about what to cover, such as how to connect the apps they design to teaching and learning, self-reflection on one's growth and struggle as an app designer, what students have learned from their peers' work and ideas, and/or documenting web resources for app design. In addition, while journal writing has been established as an approach of internal conversation for individual knowledge construction, instructors can consider encouraging students to extend it to collaborative knowledge construction by viewing and providing feedback on peers' reflective practice, which can potentially further enhance learning experiences. With the help of blogging platforms that afford participatory learning through

subscription and commenting functions, collaborative knowledge construction can be further strengthened among the members of the virtual learning community (Hsu, Ching, & Grabowski, 2013).

Engaging in External Online Learning Communities

Although students in this course were supported by the instructor and in-class learning community, educators would want to encourage students to join or participate in the established online discussion forums on mobile app design with App Inventor, such as Programming with App Inventor or App Inventor Coffee Shop. These forums consist of passionate and knowledgeable members with experience in mobile app design, and are widely available and accessible through a quick Google search. While the in-class forums offered a safe and comfortable place for sharing apps and exposing one's gap in knowledge, outside discussion forums can offer inspiration and solutions to problems not readily available in small-class environments. Additionally, extending one's learning community can help sustain learning opportunities and motivate students to continue the mobile app design journey with renewed interests and purposes.

Capturing the Questions and Need to Prepare for the Future

While students' activities, such as forum discussions and design journals, can sustain current students' in-class learning, reviewing students' activities can also help instructors capture useful information for preparing some form of "Frequently Asked Questions" and "Tips for the Week" for future students. While it is important to have students learn by solving design problems, providing organized tips can be helpful for novices when multiple technical issues can develop and become frustrating to students.

Conclusions

In this study, we presented non-programmer adult students' learning experiences and perceptions toward mobile app design using a web-based visual programming tool in an online graduate course. We found the students were positive about their online learning experiences. They also appreciated the rich peer support in this virtual learning community, which consisted of multiple online learning environments and activities that supplemented each other. In addition, they recognized the great sense of empowerment and achievement for being able to use a programming tool to create something useful that might not have existed before, or to use the tool or apps in their own daily personal or professional settings. While the students felt constrained by the inflexible interface of App Inventor on the Component Designer, and felt inconvenienced in reviewing programming blocks when their apps became too complicated, most students commented highly on its sophistication and capability to unleash their creativity and create something fully functioning, fun, and useful. Finally, we discussed instructional design implications based on students' feedback and the instructor's own reflection. We hope the study presented here showed a promising

example of teaching non-programmer adults mobile app design by integrating various components that supplemented each other to create a positive online learning experience. We also believe this study depicts the power and potential of the web-based App Inventor programming tool for motivating students to learn programming logic and encourage students to engage in creative problem solving through mobile app design activities. This study shows the educational value of mobile app design activities, and the possibility and practicality of teaching/learning mobile app design online, which can further encourage educators to explore and experiment on the potential of incorporating these learning activities in their diverse settings. However, we would like to acknowledge the small sample size of our study and remind readers to interpret our findings with caution. We would also like to encourage future research replicating our study with students of similar or different backgrounds and learning contexts to validate our findings and benefit educators and their students.

References

- Baker, A., Jensen, P., & Kolb, D. (2002). *Conversational learning: An experiential approach to knowledge creation*. London: Quorum.
- Carnegie Mellon University (2008). What is Alice? Retrieved from http://www.alice.org/index.php?page=what_is_alice/what_is_alice.
- Ferdig, R., & Trammell, K. (2004). Content delivery in the 'blogosphere.' *T.H.E. Journal*. Retrieved from <http://www.thejournal.com/articles/16626>
- Glinert, E. P. (1986). Towards "second generation" interactive, graphical programming environments. *Proceedings of 2nd IEEE Computer Society Workshop on Visual Languages*, Dallas, Texas, USA.
- Hsu, Y. -C., & Ching, Y. -H. (2011). Microblogging for strengthening a virtual learning community in an online course. *Knowledge Management & E-Learning: An International Journal*, 3(4), 375-388.
- Hsu, Y. -C., & Ching, Y. -H. (2012). Mobile microblogging: Using Twitter and mobile devices in an online course to promote learning in authentic contexts. *The International Review of Research in Open and Distance Learning*, 13(4), 211-227.
- Hsu, Y. -C., Ching, Y. -H., & Grabowski, B. (2013). Web 2.0 applications and practices for learning through collaboration. In M. Spector, D. Merrill, J. Elen & M. J. Bishop (Eds.), *Handbook of research on educational communications and technology* (4th ed.). New York, NY: Springer Academics.
- Hsu, Y. -C., Rice, K., & Dawley, L. (2012). Empowering educators with Google's Android App Inventor: An online workshop in mobile app design. *British Journal of Educational Technology*, 43(1), E1-E5. doi:10.1111/j.1467-8535.2011.01241.x
- Johnson, L., Levine, A., Smith, R., & Stone, S. (2010). *The 2010 horizon report*. Austin, Texas: The New Media Consortium.
- Johnson, L., Adams, S., & Cummins, M. (2012). *Mobile apps. The NMC horizon report: 2012 higher education edition*. Austin, Texas: The New Media Consortium.
- Ladyshevsky, R. K., & Gardner, P. (2008). Peer assisted learning and blogging: A strategy to promote reflective practice during clinical fieldwork. *Australasian Journal of Educational Technology*, 24(3), 241–257.
- Lock, J. (2002). Laying the groundwork for the development of learning communities within online courses. *The Quarterly Review of Distance Education*, 3(4), 395-408.

Lifelong Kindergarten Group (2006). About Scratch. Retrieved from
http://info.scratch.mit.edu/About_Scratch.

Margulieux, L. E., Guzdial, M., & Catrambone, R. (2012). Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. *Proceedings of the ninth annual international conference on International computing education research - ICER '12*, 71. doi:10.1145/2361276.2361291

McKelvey, N., & Curran, K (2012). Teaching Java programming using CABLE in a collaborative online environment. *International Journal of Evaluation and Research in Education*, 1(1), 33-40.

Massachusetts Institute of Technology (2012). App inventor stories. Retrieved from
<http://appinventor.mit.edu/explore/stories.html>.

Morelli, R., Lake, P., Limardo, N., & Tamotsu, E. (2011). Can Android App Inventor bring computational thinking to K-12? *Computer*. Retrieved from
http://www.hfoss.org/uploads/docs/appinventor_manuscript.pdf

Rovai, A. P. (2002a). Development of an instrument to measure classroom community. *The Internet and Higher Education*, 5, 197-211.

Rovai, A. P. (2002b). Sense of community, perceived cognitive learning, and persistence in asynchronous learning networks. *The Internet and Higher Education*, 5, 319-332.

Sia, C. L., Tan, B. C. Y., & Wei, K. K. (2002). Group polarization and computer-mediated communication: Effects of communication cues, social presence, and anonymity. *Information Systems Research*, 13(1), 70–90.

Shoffner, M. (2009). Personal attitudes and technology: Implications for pre-service teacher reflective practice. *Teacher Educational Quarterly*, 36(2), 143-161.

Shumack, K. (2010). The conversational self : Structured reflection using journal writings. *Journal of Research Practice*, 6(2), 1–22.

Turbak, F., Sandu, S., Kotsopoulos, O., Erdman, E., Davis, E., & Chadha, K. (2012). Blocks languages for creating tangible artifacts. *The Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing* (VLHCC 2012), Innsbruck, Austria. Retrieved from
<http://cs.wellesley.edu/~fturbak/pubs/VLHCC-2012-paper-turbak.pdf>.

Wang, W. (2011), *Teaching programming online*. International conference on the future of education. Retrieved from http://www.pixel-online.net/edu_future/common/download/Paper_pdf/ELE19-Wang.pdf

Wikipedia (2013a). App Store (iOS). Retrieved from
http://en.wikipedia.org/wiki/App_Store_%28iOS%29.

Wikipedia (2013b). Google Play. Retrieved from
http://en.wikipedia.org/wiki/Google_play.

Wolber, D. (2010). A blocks language for mobile phones: App Inventor for Android. In E. Canessa & M. Zennaro (Eds.), *mScience: Sensing, computing and dissemination*. Trieste, Italy : ICTP—The Abdus Salam International Centre for Theoretical Physics.

Wolber, D. (2011). App inventor and real-world motivation. *Proceedings of the 42nd ACM technical symposium on Computer science education SIGCSE 11* (p. 601). ACM Press. doi:10.1145/1953163.1953329

Wolber, D., Abelson, H., Spurtus, E., & Looney, L. (2011). *App Inventor: Create your own Android apps*. Sebastopol, CA: O'Reilly Media.

Athabasca University 

